

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное учреждение
высшего профессионального образования
"Казанский (Приволжский) федеральный университет"
Институт фундаментальной медицины и биологии



УТВЕРЖДАЮ

Проректор
по образовательной деятельности КФУ
Проф. Минзарипов Р.Г.

_____ 20__ г.

Программа дисциплины

Компьютерные технологии и в биологии. Программирование и математическое моделирование.
М1.Б.1

Направление подготовки: 020400.68 - Биология

Профиль подготовки: Биоинформатика

Квалификация выпускника: магистр

Форма обучения: очное

Язык обучения: русский

Автор(ы):

Акберова Н.И. , Тарасов Д.С.

Рецензент(ы):

Темников Д.А.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой:

Протокол заседания кафедры No ____ от " ____ " _____ 201__ г

Учебно-методическая комиссия Института фундаментальной медицины и биологии:

Протокол заседания УМК No ____ от " ____ " _____ 201__ г

Регистрационный No

Казань
2014

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) доцент, к.н. (доцент) Акберова Н.И. кафедра биохимии ИФМиБ отделение фундаментальной медицины , Natasha.Akberova@kpfu.ru ; Тарасов Д.С.

1. Цели освоения дисциплины

знакомство со способами математической формализации биологических процессов; с основными приемами моделирования сложных биологических систем и методов анализа моделей, обсуждение применения различных вычислительных схем расчета моделей; знакомство с классическими моделями в биологии и демонстрация значения математического и компьютерного моделирования для понимания природы биологических процессов и функционирования биологических систем

2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел " М1.Б.1 Общенаучный" основной образовательной программы 020400.68 Биология и относится к базовой (общепрофессиональной) части. Осваивается на 1 курсе, 2 семестр.

При изучении данного курса используются знания и навыки, полученные при изучении курсов "Информатика и программирование", "Дополнительные главы математики".

Полученные знания и навыки используются при выполнении курсовых и выпускной квалификационной работ.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-6 (профессиональные компетенции)	творчески применяет современные компьютерные технологии при сборе, обработке, анализе и передаче биологической информации;
ОК-6 (общекультурные компетенции)	способен самостоятельно приобретать информацию с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности
ПК-3 (профессиональные компетенции)	самостоятельно анализирует имеющуюся информацию, выявляет фундаментальные проблемы, ставит задачу и выполняет полевые, лабораторные биологические исследования для решения конкретных задач по специализации с использованием современной аппаратуры и вычислительных средств, демонстрирует ответственность за качество работ и научную достоверность результатов;

В результате освоения дисциплины студент:

1. должен знать:

- основные принципы функционального программирования и базовые конструкции языков функционального программирования на F#

2. должен уметь:

- разрабатывать программы средней сложности с использованием F#

3. должен владеть:

- представлениями о различных способах мышления в программировании и парадигмах программирования, о языках логического и функционального программирования, их особенностях;

-

применять навыки программирования для решения биологических задач

применять навыки программирования для решения биологических задач

применять навыки программирования для решения биологических задач

применять навыки программирования для решения биологических задач

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет зачетных(ые) единиц(ы) 144 часа(ов).

Форма промежуточного контроля дисциплины экзамен во 2 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Модуль 1. Программирование	2	1-3	0	6	0	коллоквиум
2.	Тема 2.	2	4-5	0	6	0	контрольная работа
3.	Тема 3.	2	6-8	0	6	0	контрольная работа
4.	Тема 4. Модуль 2. Математическое моделирование Математическое моделирование метаболических путей	2	1-8	0	6	0	контрольная работа
5.	Тема 5. Триггерные системы в биологии	2	8-13	0	6	0	контрольная работа

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
6.	Тема 6. Модели взаимодействующих видов. Автоколебательные процессы в биологических системах	2	13	0	6	0	контрольная работа
	Тема . Итоговая форма контроля	2		0	0	0	экзамен
	Итого			0	36	0	

4.2 Содержание дисциплины

Тема 1. Модуль 1. Программирование

практическое занятие (6 часа(ов)):

Тема 2.

практическое занятие (6 часа(ов)):

Тема 3.

практическое занятие (6 часа(ов)):

Тема 4. Модуль 2. Математическое моделирование Математическое моделирование метаболических путей

практическое занятие (6 часа(ов)):

Тема 5. Триггерные системы в биологии

практическое занятие (6 часа(ов)):

Тема 6. Модели взаимодействующих видов. Автоколебательные процессы в биологических системах

практическое занятие (6 часа(ов)):

4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Модуль 1. Программирование	2	1-3			
2.	Тема 2.	2	4-5			
3.	Тема 3.	2	6-8			
4.	Тема 4. Модуль 2. Математическое моделирование Математическое моделирование метаболических путей	2	1-8			
5.	Тема 5. Триггерные системы в биологии	2	8-13			

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
6.	Тема 6. Модели взаимодействующих видов. Автоколебательные процессы в биологических системах	2	13			
	Итого				0	

5. Образовательные технологии, включая интерактивные формы обучения

При освоении дисциплины предусматривается широкое использование активных и интерактивных форм приобретения новых знаний. В обязательном порядке должен быть обеспечен доступ студентов в Интернет. В курсе запланирована активная работа в компьютерном классе. Желательно также, чтобы студентам была предоставлена возможность удаленного доступа к ресурсам на вычислительном кластере.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Тема 1. Модуль 1. Программирование

Тема 2.

Тема 3.

Тема 4. Модуль 2. Математическое моделирование Математическое моделирование метаболических путей

Тема 5. Триггерные системы в биологии

Тема 6. Модели взаимодействующих видов. Автоколебательные процессы в биологических системах

Тема . Итоговая форма контроля

Примерные вопросы к экзамену:

Как описываются рекурсивные функции в F#?

При введении списков, как определяется функция отделения первого элемента hd?

Что такое замыкание?

Что такое мемоизация?

Какие операторы традиционно отсутствуют в функциональных языках?

Какие языки программирования являются преимущественно функциональными?

В чем отличия функционального программирования и императивного?

Что такое монада?

Какого типа выражение `async { return 10; }`?

Сколько потоков выполнения операционной системы будет порождаться при выполнении операции `Async.RunSynchronously(Async.Parallel([for i in 1..100 async { return fib(i) }]))`?

Как правильно описать двоичное дерево на F#?

В каком порядке обходятся поддеревья в инфиксном порядке обхода?

Что такое продолжение (continuation)?

Как правильно расставить скобки в выражении лямбда-исчисления, чтобы вычислить $f(g(x))$?

При нормальном порядке редукции:

Какое максимальное количество неподвижных точек может иметь функция?

При статическом контроле типов:

Алгоритм вывода типов W содержит в себе следующие основные этапы:

Как определяется наименьшая неподвижная точка непрерывной функции f в соответствии с теоремой о неподвижной точке?

Какой тип имеет функция `eval` в `Eval/Apply`-интерпретаторе?

При реализации синтаксического анализатора методом рекурсивного спуска, какой тип удобно использовать для функции `parse`:

Какие условия являются необходимыми для хвостовой рекурсии?

Какие недостатки "классической" нотации для определения функции $f(x)=2*x+1$?

Как описать повторяющиеся действия в функциональном языке?

Как определяются значения логического типа при формальном построении языка функционального программирования?

Как определяется конструкция `letrec`?

Пусть L - генератор последовательности длины n . Какова сложность операции `map f L`?

Какие основные способы борьбы со сложностью используются в функциональных программах?

Какая алгоритмическая модель лежит в основе функционального программирования?

За счет чего функциональные программы обычно содержат меньше ошибок, чем императивные?

Что означает запись `<@ fun x x*2 @>`?

Укажите правильное монадическое свойство:

Как правильно параллельно вычислить сумму 100-го и 200-го чисел Фибоначчи?

Как связаны двоичные деревья и деревья общего вида?

Как обычно представляется дерево арифметического выражения?

Как можно свести нелинейно-рекурсивную функцию к хвостовой рекурсии?

Какой самый внутренний редекс в выражении $(\lambda x.\lambda y.y)((\lambda z.z z)(\lambda z.z z))$?

Как определяют рекурсивные вычисления в лямбда-исчислении?

В динамических языках программирования:

Какое множество значений у типа $T_1 T_2$?

Неразрешимость проблемы остановки означает:

Что помещается в стек объектов S SECD-машины?

В каком представлении матриц проще реализовать операцию транспонирования?

Какова сложность проверки вхождения элемента в список длины n ?

Где может использоваться сопоставление с образцом в `F#`?

Как представляется число n в виде нумерала Черча?

Как определяется конструкция `let`?

Укажите правильное монадическое свойство:

Чему равен результат выражения `pondet { return 10; }` для монады недетерминированных вычислений?

Какие ошибки содержатся в приведенном фрагменте кода?

```
1 let ProcessImageAsync () =
```

```
2 async { let inStream = File.OpenRead(sprintf "Image%d.tmp" i)
```

```
3 let pixels = inStream.ReadAsync(numPixels)
```

```
4 let pixels' = TransformImage(pixels,i)
```

```
5 let outStream = File.OpenWrite(sprintf "Image%d.done" i)
```

```
6 do outStream.WriteAsync(pixels') }
```

Как правильно описать дерево общего вида на `F#`?

Какова сложность поиска в дереве поиска?

Какая модель вычислений служит формальной основой функционального программирования?

Как расставляются скобки в выражении $\lambda x.\lambda y.e1e2$?

Какой порядок редукции соответствует передаче параметров по значению?

Что необходимо сделать для вывода типов в некотором выражении?

Возможно ли построить автоматический верификатор произвольных функциональных программ на зацикливаемость?

Что помещается в стек возвратов D SECD-машины?

Почему контекстно-свободная грамматика удобна для разбора методом рекурсивного спуска?

Какая функция может быть использована для удаления из списка всех элементов, делящихся на 3?

Какой функции эквивалентна запись $[\text{for } x \text{ in } L \ x^*2]$?

Что такое каррирование?

Можно ли в F# изменять контекст вычисления функции внутри замыкания?

Как реализуются повторяющиеся действия в функциональных языках?

Какая алгоритмическая модель лежит в основе императивного программирования?

В чем разница между записями $\langle @ \text{ fun } x \ x^*2 \ @ \rangle$ и $\langle @@ \text{ fun } xx^*2 \ @@ \rangle$?

Что такое дерево поиска?

Какие есть основные модели вычислений?

Какое из приведенных ниже преобразований является примером альфа-редукции?

Какой будет следующий шаг при аппликативном порядке редукции выражения $(\lambda x.x+x)(2+3)$?

Как определяется множество значений для типа, описанного как $\text{type tree} = \{\text{nil}\} + \text{int} \times \text{tree} \times \text{tree}$?

Что такое домены?

Какой тип имеет функция filter?

Какая рекурсия называется линейной?

Почему функциональное программирование сейчас представляет повышенный интерес для изучения?

В каком порядке обходятся поддеревья в префиксном порядке обхода?

Какой будет наиболее общий тип для функции $\text{tl}: \text{let } \text{tl } x::t = t$?

Почему следует стараться использовать хвостовую рекурсию?

В каком представлении эффективнее хранить разреженные матрицы?

Какой принцип построения функциональных программ?

Почему функциональные программы не содержат побочных эффектов?

Система вывода типов (Хиндли-Милнера) - это:

Что такое пропозициональная семантика языка программирования?

Для реализации ленивого Eval/Apply-интерпретатора необходимо, в частности:

Какой будет результат сопоставления $\text{let } x::y::z = [1;2]$?

В чем разница между записями $\langle @ \text{ fun } x \ x^*2 \ @ \rangle$ и $\text{fun } x \ \langle @ \ x^*2 \ @ \rangle$?

Как записать выражение для последующего параллельного вычисления значения функции fib?

Какое из приведенных ниже преобразований является примером бета-редукции?

Какому способу передачи параметров соответствует аппликативный порядок редукции?

Зачем в F# необходимо статически ассоциировать типы с именами?

Что такое денотационная семантика языка программирования?

Как определяется семантика рекурсивных функций?

Какова сложность добавления элемента на первое место списка длины n?

Какие основные операции в чистом λ -исчислении?

- Какой тип у каррированной функции сложения целых чисел?
- Возможно ли в F# использовать ленивые вычисления?
- Какая операция повышает порядок функции?
- Какой самый внешний редекс в выражении $(\lambda x.\lambda y.y)((\lambda z.z z)(\lambda z.z z))$?
- Какое множество значений у прямой суммы T_1+T_2 ?
- Какая функция, определенная на домене D, обязательно имеет неподвижную точку?
- Почему проще формулировать и доказывать корректность программ на чистом функциональном языке?
- При вычислении длины списка n с помощью хвостовой рекурсии, сколько памяти выделяется в стеке?
- Как расставляются скобки в выражении $f \sin x^2$?
- Какой будет следующий шаг при нормальном порядке редукции выражения $(\lambda x.x+x)(2+3)$?
- Что является следствием теоремы Черча-Россера?
- Какими способами можно создать список четных чисел от 1 до 10?
- За счет чего функциональные программы содержат меньше ошибок?
- Что является следствием теоремы стандартизации?
- Как можно разделить голову и хвост списка?
- С помощью какой функции можно удалить из списка все элементы, стоящие на четных позициях?
- В чем разница между конструкциями fun и function в F#?
- Что такое операционная семантика языка программирования?
- В чем разница между конструкциями fun и function в F#?
- Eval/Apply-интерпретатор состоит из
- Пусть геометрическое преобразование определяется функцией трансляции координат $int*int \rightarrow int*int$. Мы хотим определить функцию сдвига $translate : int*int \rightarrow int*int$, которая возвращалась бы замыкание. Как это сделать?
- Разреженная матрица размерности $n \times n$ с m ненулевыми элементами представляется в виде функции $int*int \rightarrow float$. Какова будет сложность операции умножения всех элементов матрицы на 2?
- Какого типа выражение $\langle @@ (1+2)*3 @@ \rangle$?
- Какой будет результат выполнения `let x::y = [1;2;3;4] ?`
- Какие комбинаторы образуют наименьший базис?
- Какая операция понижает порядок функции?
- Какие базовые операции чистого лямбда-исчисления?
- В каком порядке обходятся поддеревья в постфиксном порядке обхода?
- В чем преимущества лямбда-исчисления как модели вычислений?
- Какие из преобразований надо применить, чтобы редуцировать $(\lambda x.\sin x) 0 0$?
- Какой тип имеет функция `map`?
- Какова сложность добавления элемента в конец списка длины n?
- Какая операция применяет функцию к аргументу?

7.1. Основная литература:

Леск А. Введение в биоинформатику М.: БИНОМ.Лаборатория знаний, 2009. - 318с. 1 экз

7.2. Дополнительная литература:

Фролова, Людмила Леонидовна. Базы данных нуклеотидных последовательностей Genbank/EMBL/DDBJ. Аннотация гена arcA E.coli K12: учебное пособие по курсу "Биоинформатика" / Л. Л. Фролова, А. Я. Хидиятуллина, А. С. Кузьмина; Казан. гос. ун-т.Казань: Изд-во Казан. гос. ун-та, 2007.45 с.: ил.; 21.Библиогр.: с. 44, 50.

7.3. Интернет-ресурсы:

8. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Освоение дисциплины "Компьютерные технологии и в биологии. Программирование и математическое моделирование." предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB, audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудование имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 020400.68 "Биология" и магистерской программе Биоинформатика .

Автор(ы):

Акберова Н.И. _____

Тарасов Д.С. _____

"__" _____ 201__ г.

Рецензент(ы):

Темников Д.А. _____

"__" _____ 201__ г.